

# MATHEMATICS FOR SOFTWARE ENGINEERING

## Mathematics for Software Engineering

**Authors:** Richard Brooks & Eduard Fekete

**Date:** May, 2025

**Version:** 2.0

# Contents

<b>Chapter 1 Descriptive Statistics</b>	<b>1</b>
1.1 Introduction to Descriptive Statistics . . . . .	1
1.2 Describing Data Sets . . . . .	3
1.3 Summarizing Data . . . . .	8
1.4 Understanding Data Distributions . . . . .	14
<b>Appendix A Important Concepts</b>	<b>20</b>

# Chapter 1 Descriptive Statistics

In software engineering, we constantly deal with data: performance metrics, user behavior patterns, system response times, and code complexity measures. To make sense of this information and extract meaningful insights, we need tools to summarize, organize, and visualize data. Descriptive statistics provides these essential tools, allowing us to understand the characteristics of our data without making inferences about larger populations.

Consider a software development team analyzing the performance of their web application. They collect response times for 1000 user requests and find values ranging from 50 milliseconds to 3.2 seconds. Without proper analysis, this raw data is overwhelming and uninformative. However, by applying descriptive statistics, they can determine that the average response time is 450 milliseconds, that 75% of requests complete within 600 milliseconds, and that there are a few unusually slow requests that might indicate performance issues.

This chapter introduces the fundamental concepts of descriptive statistics, focusing on measures that help us understand the central tendencies, variability, and distribution patterns in our data. We will explore how these techniques apply specifically to software engineering contexts, from analyzing algorithm performance to understanding user behavior patterns.

## 1.1 Introduction to Descriptive Statistics

Sometimes statistical work begins with existing data, such as precipitation records, unemployment rates, or GDP figures, which we then summarize and analyze. In other situations, data must be generated through an experiment or study. For example, to compare two teaching methods in an introductory programming course, an instructor might randomly divide students into two groups, apply a different method to each, and then compare test scores. Random assignment is essential: it ensures that differences between groups are not due to preexisting factors such as aptitude, but rather to the teaching method itself.

After the data are collected, they are summarized and visualized—for example, by reporting the average score for each group. This process is the essence of **descriptive statistics**: organising, presenting, and describing data in a meaningful way.

### Population versus Sample

In statistical analysis, we distinguish between two key concepts:

#### Definition 1.1 (Population)

A **population** is the complete set of all possible observations or measurements of interest in a particular study. In software engineering contexts, this might include all possible execution times of an algorithm, all user sessions on a website, or all lines of code in a project.



**Definition 1.2 (Sample)**

A **sample** is a subset of the population that we actually observe or measure. Due to practical constraints, we often work with samples rather than entire populations.



In the teaching-method example, the students in the classroom form only a *sample* of the larger *population* of potential learners. Descriptive statistics helps us understand the sample itself, while inferential methods later allow us to use that sample to draw conclusions about the broader population.

**Example 1.1** Software Performance Analysis

Consider a web application serving millions of users daily. The **population** would be the response times for all possible user requests. However, due to computational and storage limitations, we might only collect response times for a **sample** of 10,000 requests per day. This sample should be representative of the population to draw meaningful conclusions.

**Types of Data**

Data can be classified into different types, each requiring different statistical approaches:

**Definition 1.3 (Qualitative Data)**

**Qualitative data** (also called categorical data) consists of non-numerical information that can be categorized. Examples include programming languages used in a project, user satisfaction ratings (satisfied/neutral/dissatisfied), or bug severity levels (critical/high/medium/low).



The essential difference between **qualitative** and **quantitative** data lies in what they represent and how they are analyzed. Qualitative data refers to categories or attributes that cannot be expressed meaningfully as numbers. Such data are used to classify or label elements, for example, programming language or bug severity. Quantitative data, on the other hand, consists of numerical values that represent counts or measurements, such as response times or number of users. Because these values are numerical, they permit mathematical operations and statistical analysis. Recognizing the distinction between qualitative and quantitative data is fundamental, as it guides the choice of statistical methods and the types of visualizations that are appropriate for analysis.

**Definition 1.4 (Quantitative Data)**

**Quantitative data** consists of numerical measurements that can be ordered and subjected to mathematical operations. This can be further divided into:

- **Discrete:** Countable values (number of bugs, lines of code, user sessions)
- **Continuous:** Measurable values that can take any value within a range (response times, memory usage, CPU utilization)

**Example 1.2**

- **Qualitative:** Programming language (Python, Java, C++), deployment environment (development, staging, production)
- **Quantitative Discrete:** Number of commits per day, lines of code, number of test cases

- **Quantitative Continuous:** Response time in milliseconds, memory usage in MB, CPU utilization percentage

### Outliers and Extreme Data

In any dataset, most values tend to cluster around a central region, but occasionally, we encounter values that are much higher or lower than the rest. These are called **outliers** or **extreme values**. Outliers can arise for many reasons: measurement errors, unusual but valid events, or natural variability in the data.

For example, if most web requests complete in under 400 milliseconds, but one request takes 1200 milliseconds, that 1200ms value is an outlier. Outliers can have a strong influence on summary statistics like the mean, making the data appear more variable or shifting the average away from where most values lie.

It's important to look for outliers when analyzing data, as they can signal interesting phenomena (such as a rare performance bottleneck), data entry mistakes, or the need for further investigation. While there are formal methods to detect outliers, which we will discuss later, often a simple plot or a scan of the sorted data is enough to spot values that "stand out" from the rest.

In practice, understanding the context is key: sometimes outliers are errors to be corrected or removed, but other times they are the most important part of the story.

## 1.2 Describing Data Sets

When presenting numerical results, clarity and brevity are essential. Tables and graphs are particularly effective, as they allow the reader to quickly grasp the main characteristics of a dataset. Visual and tabular summaries often highlight important aspects such as the overall range, how tightly the values are concentrated, and whether the data appear symmetric or skewed.

### Frequency Tables and Graphs

The first step in data analysis is often to organize raw data into a frequency distribution, which shows how often each value (or range of values) occurs.

#### Definition 1.5 (Frequency Distribution)

A **frequency distribution** is a table that shows the frequency (count) of each value or class of values in a dataset. It can be presented as:

- **Absolute frequency:** The actual count of occurrences
- **Relative frequency:** The proportion of total observations
- **Cumulative frequency:** The running total of frequencies



**Example 1.3** A dataset with a relatively small number of distinct values can be conveniently presented in a frequency table. For instance, consider a dataset consisting of the starting monthly salaries (to the nearest thousand Danish Kroner) of 42 recently graduated students with B.S. degrees in software engineering.

The raw data is as follows:

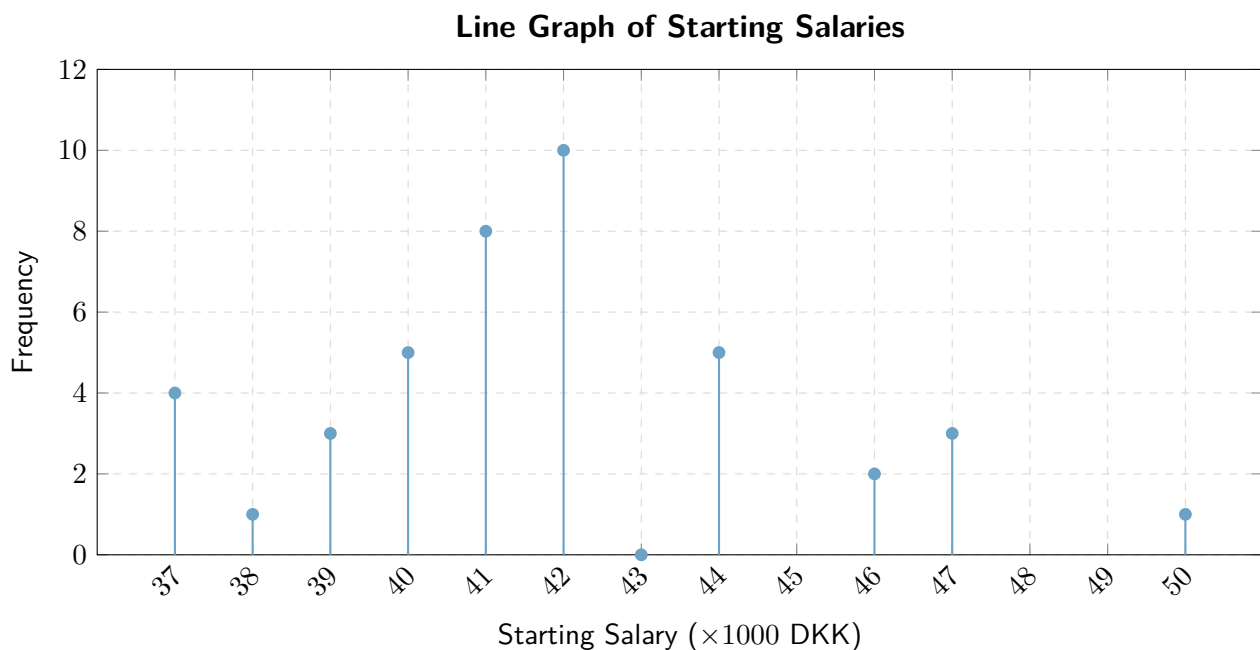
37, 37, 37, 37, 38, 39, 39, 39, 40, 40, 40, 40, 40, 41, 41, 41, 41, 41, 41, 41,  
42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 44, 44, 44, 44, 44, 46, 46, 47, 47, 47, 50

From this data, we can construct a frequency table. **Table 1.1** tells us, among other things, that the lowest starting salary of 37,000 DKK was received by four of the graduates, whereas the highest salary of 50,000 DKK was received by a single student. The most common starting salary was 42,000 DKK, received by 10 of the students.

Salary ( $\times 1000$ DKK)	Frequency
37	4
38	1
39	3
40	5
41	8
42	10
43	0
44	5
46	2
47	3
50	1

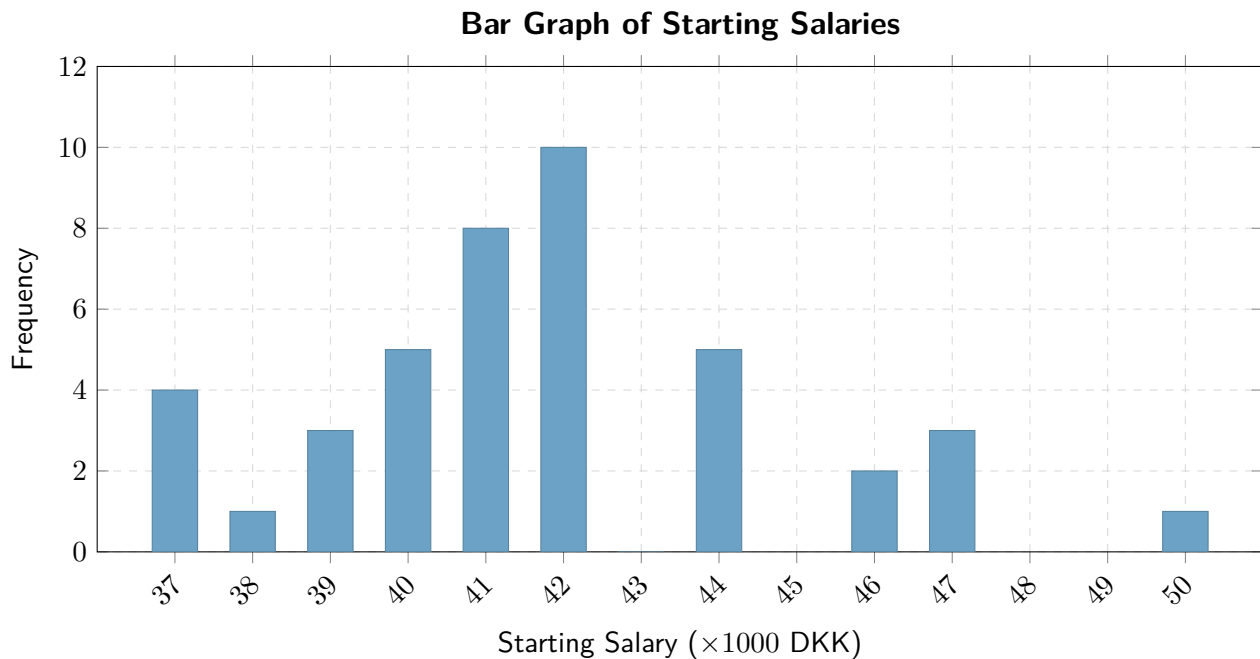
**Table 1.1:** Frequency Distribution of Monthly Starting Salaries

Data from a frequency table can be graphically represented by a line graph that plots the distinct data values on the horizontal axis and indicates their frequencies by the heights of vertical lines. A line graph of the data presented in **Table 1.1** is shown in **Figure 1.1**.



**Figure 1.1:** A line graph showing the frequency of different starting salaries.

When the lines in a line graph are given added thickness, the graph is called a bar graph. **Figure 1.2** shows a bar graph for the same salary data.



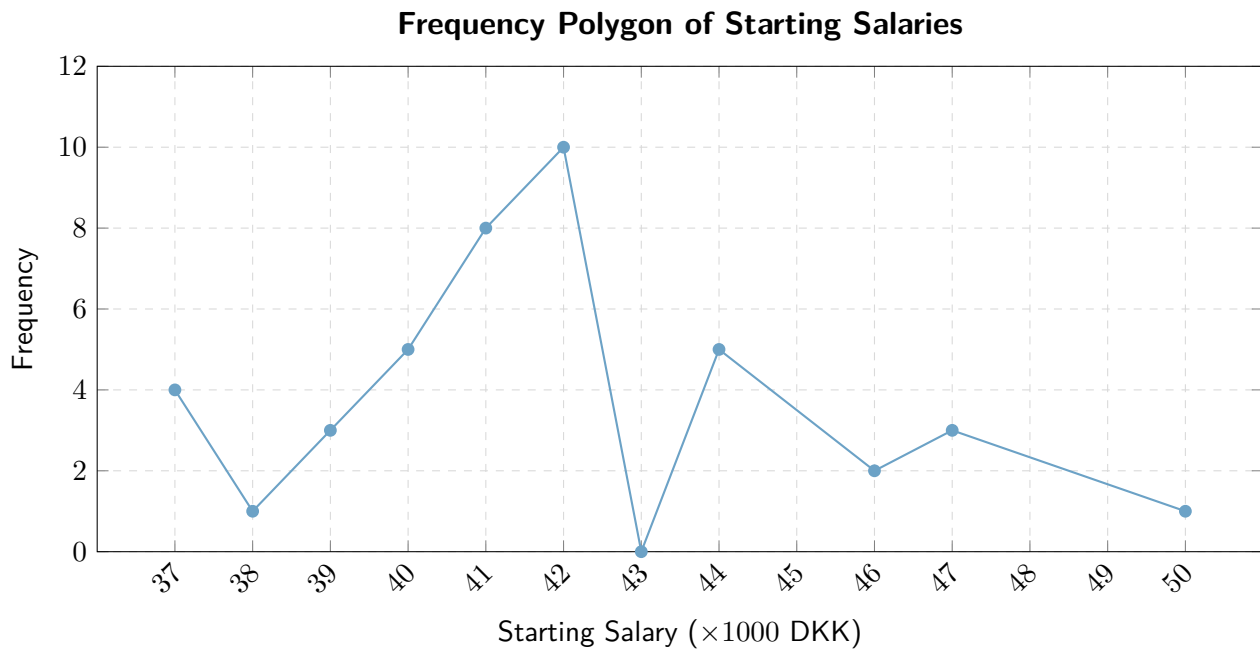
**Figure 1.2:** A bar graph of the salary data.

Another type of graph used to represent a frequency table is the frequency polygon, which plots the frequencies of the different data values on the vertical axis, and then connects the plotted points with straight lines. **Figure 1.3** shows a frequency polygon for the same salary data.

Sometimes we prefer the relative frequency over the absolute frequency. This is particularly useful when dealing with large datasets and when you want to compare values from datasets of different sizes. An extension of the relative frequency is the cumulative relative frequency, which is the running total of the relative frequencies. **Table 1.2** shows the frequency, relative frequency, and cumulative relative frequency for the salary data.

Salary	Frequency	Relative Freq.	Cumulative Rel. Freq.
37	4	$4/42 \approx 0.095$	$4/42 \approx 0.095$
38	1	$1/42 \approx 0.024$	$5/42 \approx 0.119$
39	3	$3/42 \approx 0.071$	$8/42 \approx 0.190$
40	5	$5/42 \approx 0.119$	$13/42 \approx 0.310$
41	8	$8/42 \approx 0.190$	$21/42 = 0.500$
42	10	$10/42 \approx 0.238$	$31/42 \approx 0.738$
44	5	$5/42 \approx 0.119$	$36/42 \approx 0.857$
46	2	$2/42 \approx 0.048$	$38/42 \approx 0.905$
47	3	$3/42 \approx 0.071$	$41/42 \approx 0.976$
50	1	$1/42 \approx 0.024$	$42/42 = 1.000$

**Table 1.2:** Frequency, Relative Frequency, and Cumulative Relative Frequency of Monthly Starting Salaries.



**Figure 1.3:** A frequency polygon of the salary data, connecting the points from the frequency table.

### Grouped Data and Histograms

As we saw in [section 1.2](#), line and bar graphs are useful for showing frequencies of data values. When a dataset contains a large number of distinct values, such as the lamp lifetimes in [Table 1.3](#), creating a frequency table for each individual value becomes impractical. A more effective approach is to group the data into a set of **class intervals**, which are also commonly referred to as **bins**.

The choice of how many bins to use involves a trade-off. If there are too few bins, we risk losing important information by grouping too many distinct values together. If there are too many bins, the frequencies within each bin may become too small to reveal any clear pattern in the data's distribution.

While the optimal number of bins is a subjective choice that depends on the dataset, a selection of 5 to 10 bins is typical. It is also customary, though not required, to use bins of equal width, as this often makes interpretation easier.

The endpoints of a bin are called its **class boundaries**. We will adopt the common **left-end inclusion convention**, which specifies that a bin contains its left-end boundary but not its right-end one. For example, the bin 700–800 would contain any value  $x$  such that  $700 \leq x < 800$ .

Following this approach, the raw data from [Table 1.3](#) is summarized in [Table 1.4](#) using bins of length 100, starting at 500.

A bar graph that plots the frequencies of data grouped into bins is called a **histogram**. A key feature of a histogram is that the bars are placed directly adjacent to one another, reflecting the continuous nature of the bins. The vertical axis can represent either the absolute frequency (the count of values in each interval) or the relative frequency. The former is called a **frequency histogram**, while the latter is a **relative frequency histogram**. [Figure 1.4](#) shows a frequency histogram for the lamp lifetime data.

Finally, we are sometimes interested in plotting the cumulative frequency or, more commonly, the cumulative

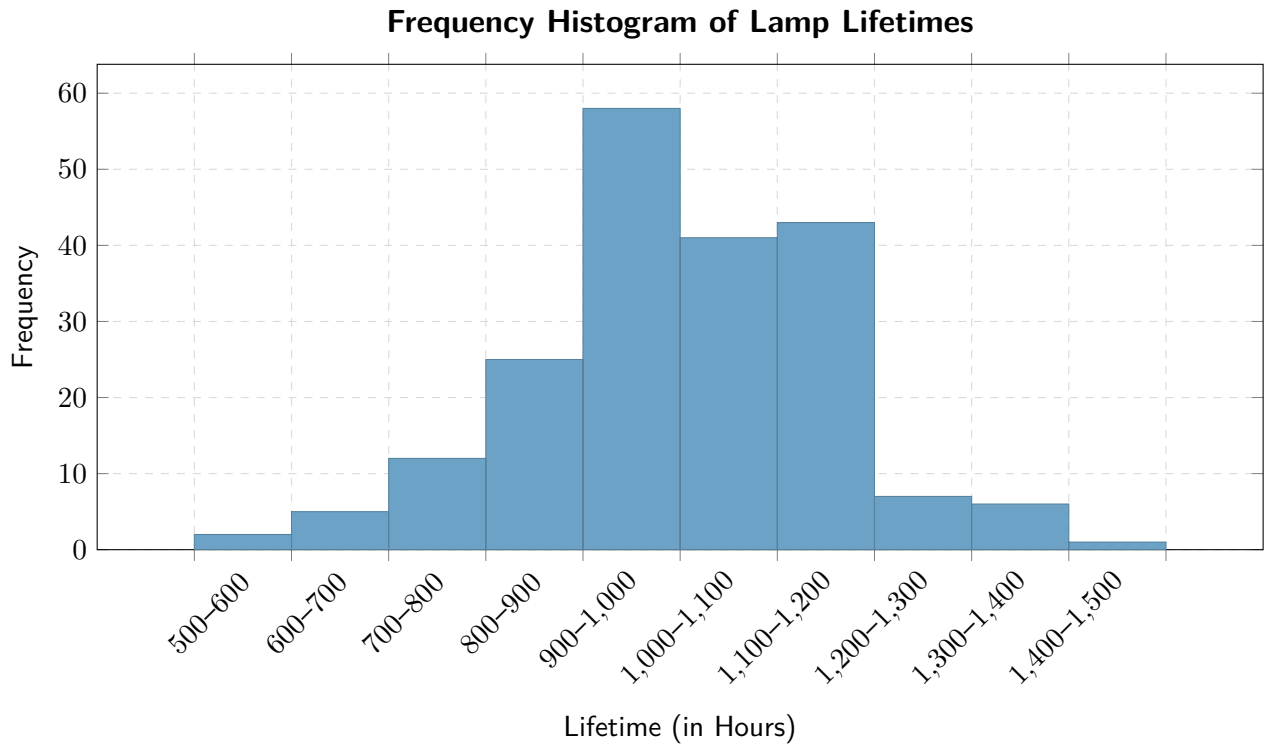


Item Lifetimes (in Hours)									
1067	919	1196	785	1126	936	918	1156	920	948
855	1092	1162	1170	929	950	905	972	1035	1045
1157	1195	1195	1340	1122	938	970	1237	956	1102
1022	978	832	1009	1157	1151	1009	765	958	902
923	1333	811	1217	1085	896	958	1311	1037	702
521	933	928	1153	946	858	1071	1069	830	1063
930	807	954	1063	1002	909	1077	1021	1062	1157
999	932	1035	944	1049	940	1122	1115	833	1320
901	1324	818	1250	1203	1078	890	1303	1011	1102
996	780	900	1106	704	621	854	1178	1138	951
1187	1067	1118	1037	958	760	1101	949	992	966
824	653	980	935	878	934	910	1058	730	980
844	814	1103	1000	788	1143	935	1069	1170	1067
1037	1151	863	990	1035	1112	931	970	932	904
1026	1147	883	867	990	1258	1192	922	1150	1091
1039	1083	1040	1289	699	1083	880	1029	658	912
1023	984	856	924	801	1122	1292	1116	880	1173
1134	932	938	1078	1180	1106	1184	954	824	529
998	996	1133	765	775	1105	1081	1171	705	1425
610	916	1001	895	709	860	1110	1149	972	1002

**Table 1.3:** Life in Hours of 200 Incandescent Lamps.

Class Interval	Frequency	Relative Freq.	Cumulative Rel. Freq.
500–600	2	0.010	0.010
600–700	5	0.025	0.035
700–800	12	0.060	0.095
800–900	25	0.125	0.220
900–1000	58	0.290	0.510
1000–1100	41	0.205	0.715
1100–1200	43	0.215	0.930
1200–1300	7	0.035	0.965
1300–1400	6	0.030	0.995
1400–1500	1	0.005	1.000

**Table 1.4:** Frequency Distribution for the Lifetimes of 200 Incandescent Lamps.



**Figure 1.4:** A frequency histogram for the lamp lifetime data. The adjacent bars show the distribution of data across continuous intervals.

relative frequency. Such a graph, often called an **ogive**, shows the number or proportion of data points that fall at or below a certain value.

On an ogive, each point on the horizontal axis represents a data value, while the corresponding point on the vertical axis shows the proportion of the data that is less than or equal to that value. For example, **Figure 1.5** presents the cumulative relative frequency plot for the lamp lifetime data. From this graph, we can quickly determine that approximately 46% of the lamps had a lifetime of less than 1000 hours, about 93% had a lifetime of less than 1200 hours, and 100% of the lamps failed by 1500 hours.

## 1.3 Summarizing Data

Having seen how to visually and numerically summarize data distributions, we now focus on two fundamental aspects of describing a dataset: its *center* (or typical value) and its *dispersion* (or variability).

### Measures of Central Tendency

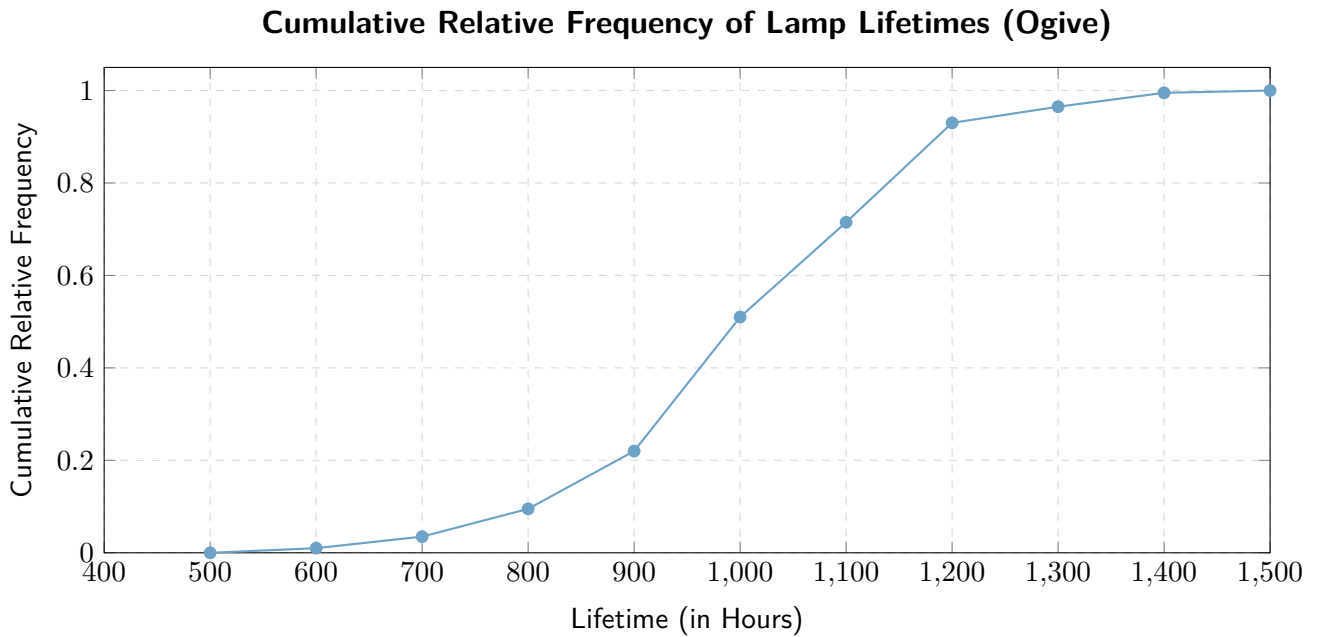
The **mean**, often called the average, is the most commonly used measure of central tendency. **Example 1.4** shows how to calculate the mean of a dataset.

#### Definition 1.6 (Arithmetic Mean)

For a dataset with  $n$  values  $x_1, x_2, \dots, x_n$ , the **arithmetic mean** is defined as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n}$$





**Figure 1.5:** An ogive showing the cumulative relative frequency for the lamp lifetime data.

#### Example 1.4 Response Time Analysis

A web application recorded the following response times (in milliseconds) for 10 requests:

{120, 150, 180, 200, 220, 250, 280, 300, 350, 400}

The mean response time is:

$$\bar{x} = \frac{120 + 150 + 180 + 200 + 220 + 250 + 280 + 300 + 350 + 400}{10} = \frac{2450}{10} = 245 \text{ milliseconds}$$

The mean incorporates every value in a dataset, which makes it a powerful measure of central tendency. It is most appropriate when the data values are of similar magnitude and when the distribution is reasonably symmetric without extreme outliers. Under these conditions, the mean provides a reliable summary of the dataset.

For example, in the analysis of web performance, the mean response time offers an overall indication of how quickly a system responds on average. However, if a few response times are much slower than the rest, the mean may give a misleading impression of the typical user experience.

Beyond its role as a summary statistic, the mean also serves as a foundation for many statistical methods. In particular, it is central to the definitions of variance and standard deviation, which measure the dispersion of data around the mean.

**Remark:** As mentioned earlier, and as we will revisit, the mean is highly sensitive to extreme values or outliers. Even a single unusually large or small observation can shift the mean considerably, reducing its ability to reflect the typical value of the dataset. It is therefore important to consider the distribution of the data before relying solely on the mean to describe the center of a dataset.

The **median** is the middle value when data is arranged in ascending order. It is less sensitive to outliers than the mean.

**Definition 1.7 (Median)**

For a dataset with  $n$  values arranged in ascending order:

- If  $n$  is odd, the median is the middle value:  

$$x_{\frac{n+1}{2}}$$
- If  $n$  is even, the median is the average of the two middle values:  

$$\frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}$$

**Example 1.5** Median Response Time

Using the same response time data from the previous example:

{120, 150, 180, 200, 220, 250, 280, 300, 350, 400}

Since  $n = 10$  (even), the median is the average of the 5th and 6th values:

$$\text{Median} = \frac{220 + 250}{2} = 235 \text{ milliseconds}$$

**Example 1.6** Median with Outlier

Consider the same data with an extreme value:

{120, 150, 180, 200, 220, 250, 280, 300, 350, 2000}

The mean becomes:  $\bar{x} = \frac{4050}{10} = 405$  milliseconds (significantly affected by the outlier)

The median remains:  $\frac{220+250}{2} = 235$  milliseconds (unaffected by the outlier)

Because the median divides a dataset into two equal halves, it serves as a robust measure of central tendency, particularly when the data contain outliers or are skewed. Unlike the mean, which can be drawn toward extreme values, the median remains largely unaffected, offering a more reliable indication of the “typical” value in such situations.

The **mode** is the most frequently occurring value in a dataset.

**Definition 1.8 (Mode)**

The **mode** is the value that appears most frequently in a dataset. A dataset can have:

- **No mode:** If all values appear with equal frequency
- **One mode (unimodal):** If one value appears most frequently
- **Multiple modes (multimodal):** If two or more values tie for the highest frequency

**Example 1.7** A software project has the following bug severity levels:

{High, Medium, Low, High, Critical, Medium, High, Low, High, Medium}

Counting frequencies:

- High: 4 occurrences
- Medium: 3 occurrences
- Low: 2 occurrences

- Critical: 1 occurrence

The mode is "High" since it appears most frequently.

The mode is especially useful when analyzing categorical or discrete data, where calculating a mean or median may not make sense. For example, in software engineering, the mode can help identify the most common error code returned by an API, the most frequently used programming language in a codebase, or the most reported type of bug in an issue tracker.

### When to Use Each Measure

Measure	Best For	Limitations
Mean	Continuous data, symmetric distributions	Sensitive to outliers
Median	Data with outliers, skewed distributions	Less informative for symmetric data
Mode	Categorical data, discrete data	May not exist or be unique

**Table 1.5:** Comparison of Central Tendency Measures

### Measures of Dispersion

While measures of central tendency tell us about the typical value, measures of dispersion describe how spread out the data is. Understanding variability is crucial in software engineering for assessing consistency and reliability.

The **range** is the simplest measure of dispersion.

#### Definition 1.9 (Range)

The **range** of a dataset is the difference between the maximum and minimum values:

$$\text{Range} = x_{\max} - x_{\min}$$



#### Example 1.8

For the response time data: {120, 150, 180, 200, 220, 250, 280, 300, 350, 400}

$$\text{Range} = 400 - 120 = 280 \text{ milliseconds}$$

**Remark:** The range is sensitive to outliers and doesn't provide information about the distribution of values between the extremes.

The **variance** and **standard deviation** are more sophisticated measures that consider all data points.

#### Definition 1.10 (Sample Variance and Standard Deviation)

For a sample with  $n$  values, the **sample variance**, denoted by  $s^2$ , is defined as:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

The **sample standard deviation**, denoted by  $s$ , is the square root of the variance:

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$



While the mean identifies the center of a dataset, the **variance** and **standard deviation** measure its variability or spread. A small standard deviation indicates that data points are clustered tightly around the mean, whereas a large standard deviation signifies they are more spread out.

The variance is the average of the squared deviations from the mean. Squaring each deviation serves two purposes: it ensures all values are positive and it gives greater weight to larger, more significant deviations. However, this calculation leaves the variance in squared units (e.g., milliseconds squared), which are not intuitive to interpret.

To solve this, we use the standard deviation, which is simply the square root of the variance. This crucial step returns the measure to the original units of the data (e.g., milliseconds), making it directly interpretable. In short, the standard deviation represents the typical distance of a data point from the mean.

#### Example 1.9 Calculating Standard Deviation

For the response time data:  $\{120, 150, 180, 200, 220, 250, 280, 300, 350, 400\}$  with  $\bar{x} = 245$ :

$$\begin{aligned} s^2 &= \frac{1}{9} [(120 - 245)^2 + (150 - 245)^2 + \cdots + (400 - 245)^2] \\ &= \frac{1}{9} [(-125)^2 + (-95)^2 + \cdots + (155)^2] \\ &= \frac{1}{9} [15625 + 9025 + \cdots + 24025] \\ &= \frac{1}{9} \times 82250 \approx 9138.89 \\ s &= \sqrt{9138.89} \approx 95.6 \text{ milliseconds} \end{aligned}$$

The **coefficient of variation** allows comparison of variability across datasets with different scales.

#### Definition 1.11 (Coefficient of Variation)

The **coefficient of variation** is the ratio of standard deviation to the mean:

$$CV = \frac{s}{\bar{x}} \times 100\%$$



#### Example 1.10 Comparing Variability Consider two systems:

- System A: Mean response time = 100ms, Standard deviation = 20ms
- System B: Mean response time = 500ms, Standard deviation = 100ms

$$\begin{aligned} CV_A &= \frac{20}{100} \times 100\% = 20\% \\ CV_B &= \frac{100}{500} \times 100\% = 20\% \end{aligned}$$

Both systems have the same relative variability (20%), despite different absolute standard deviations.

To introduce a measure of spread that is less sensitive to outliers, we first need to understand measures of position, which describe the relative standing of a data point within a dataset.

#### Definition 1.12 (Percentile)

The  $p$ -th **percentile** is the value below which  $p\%$  of the data falls.



For example, if a response time is at the 95th percentile, it means that 95% of all response times are faster than this one.

#### Example 1.11 Percentile Calculation

For the response time data, to find the 90th percentile:

1. Sort data: {120, 150, 180, 200, 220, 250, 280, 300, 350, 400}
2. Position:  $0.9 \times 10 = 9$ th position
3. 90th percentile = 350ms

Quartiles are specific, widely used percentiles that divide the data into four equal parts.

#### Definition 1.13 (Quartiles)

Quartiles divide an ordered dataset into four equal parts.

- **Q1 (First Quartile)**: The 25th percentile. 25% of the data is less than this value.
- **Q2 (Second Quartile)**: The 50th percentile. This is the **median** of the dataset.
- **Q3 (Third Quartile)**: The 75th percentile. 75% of the data is less than this value.
- **Q4 (Fourth Quartile)**: The 100th percentile. 100% of the data is less than this value.



#### Example 1.12 Calculating Quartiles

For the sorted response time data: {120, 150, 180, 200, 220, |250, 280, 300, 350, 400}

- The lower half is {120, 150, 180, 200, 220}. The median of this half is  $Q1 = 180$  ms.
- The upper half is {250, 280, 300, 350, 400}. The median of this half is  $Q3 = 300$  ms.

Percentiles divide data into 100 equal parts, while quartiles divide data into four equal parts.

#### Interquartile Range (IQR)

The interquartile range is a robust measure of dispersion that is not affected by outliers.

#### Definition 1.14 (Interquartile Range)

The **interquartile range (IQR)** measures the spread of the middle 50% of the data and is defined as

$$IQR = Q_3 - Q_1,$$

where  $Q_1$  and  $Q_3$  are the first and third quartiles. A larger IQR indicates that the central portion of the data is more widely dispersed, while a smaller IQR indicates that it is more tightly clustered.



#### Example 1.13 IQR Calculation

For the response time data: {120, 150, 180, 200, 220, 250, 280, 300, 350, 400}

- Lower half: {120, 150, 180, 200, 220}  $\rightarrow Q1 = 180$
- Upper half: {250, 280, 300, 350, 400}  $\rightarrow Q3 = 300$
- $IQR = 300 - 180 = 120$  milliseconds

## Summary Statistics

Before we conclude this chapter, we will review the summary statistics that we have covered in this section. Let us consider the response time data again, but now in the form of a table of 1000 values. [Table 1.6](#) shows the complete statistical summary of the data.

**Example 1.14 Complete Statistical Summary** For a dataset of 1000 web response times:

Measure	Value
Count	1000
Mean	245.3 ms
Median	238.0 ms
Mode	220.0 ms
Standard Deviation	95.6 ms
Variance	9138.9 ms <sup>2</sup>
Minimum	45.0 ms
Maximum	1200.0 ms
Range	1155.0 ms
Q1	180.0 ms
Q3	300.0 ms
IQR	120.0 ms
Skewness	1.2 (positive skew)
Kurtosis	2.8 (leptokurtic)

**Table 1.6:** Complete Statistical Summary

This table provides a complete picture of the data and is often a first step in the analysis of a dataset. Note that the skewness and kurtosis have not been discussed yet but will be explained in the next section. We have chosen to include them here for completeness as they are often used in the analysis of a dataset.

## 1.4 Understanding Data Distributions

Once we have visualized our data with histograms and box plots, we can begin to analyze the *shape* of the distribution. The shape tells us about the underlying patterns in our data and helps us choose the right statistical tools. In software engineering, understanding the distribution of metrics like response times or error rates is critical for setting performance baselines and detecting anomalies. First we review the concept of outliers.



## Outlier Detection

As mentioned in [section 1.1](#), outliers are data points that are significantly different from the rest of the data. They can arise due to measurement errors, data entry errors, or genuine variability in the data. Detecting outliers is important because they can distort statistical analyses.

A common method for identifying outliers is to use the interquartile range (IQR):

- **Mild outliers:** Any data point less than  $Q_1 - 1.5 \times IQR$  or greater than  $Q_3 + 1.5 \times IQR$ .
- **Extreme outliers:** Any data point less than  $Q_1 - 3 \times IQR$  or greater than  $Q_3 + 3 \times IQR$ .

### Example 1.15 Detecting Outliers

Given the response time data: {120, 150, 180, 200, 220, 250, 280, 300, 350, 400}

- $Q_1 = 180$ ,  $Q_3 = 300$ ,  $IQR = 120$
- Lower fence:  $Q_1 - 1.5 \times IQR = 180 - 180 = 0$
- Upper fence:  $Q_3 + 1.5 \times IQR = 300 + 180 = 480$

All data points are between 0 and 480, so there are no mild outliers in this dataset.

### Example 1.16

Suppose we have the following set of response times (in milliseconds) for a web application:

{120, 150, 180, 200, 220, 250, 280, 300, 350, 400, 1200}.

Let's identify if there is an upper outlier:

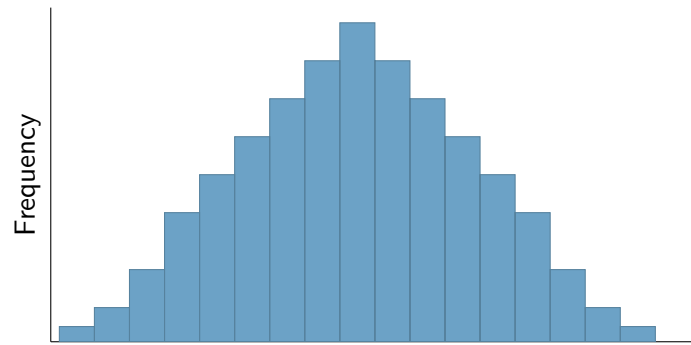
- Sorted data: 120, 150, 180, 200, 220, 250, 280, 300, 350, 400, 1200
- $n = 11$
- Median ( $Q_2$ ): 250 (6th value)
- Lower half: 120, 150, 180, 200, 220  $\rightarrow Q_1 = 180$
- Upper half: 280, 300, 350, 400, 1200  $\rightarrow Q_3 = 350$
- $IQR = Q_3 - Q_1 = 350 - 180 = 170$
- Upper fence:  $Q_3 + 1.5 \times IQR = 350 + 1.5 \times 170 = 350 + 255 = 605$

The value 1200 is greater than 605, so it is an upper outlier.

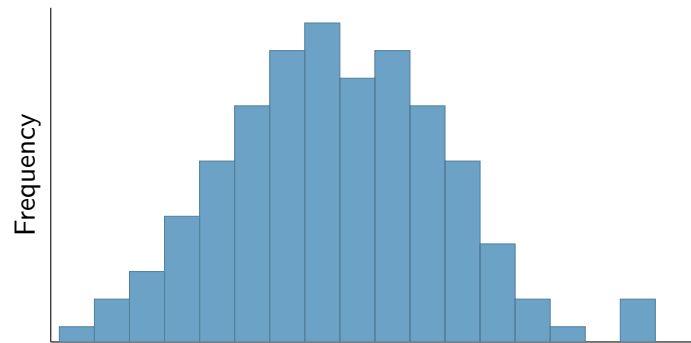
## Normal Data Sets

Many large data sets yield histograms with a distinctive overall shape: the frequencies peak near the sample median and decrease symmetrically on both sides, forming a characteristic **bell-shaped curve**. Data sets exhibiting this property are called **normal**, and their histograms are referred to as **normal histograms** (see [Figure 1.6](#)).

In software systems, perfectly normal distributions are rare, but some metrics can be **approximately normal**. For example, the latency of a highly optimized, internal microservice that performs a consistent task might be approximately normal. A histogram of an approximately normal data set looks like [Figure 1.7](#).



**Figure 1.6:** A histogram of a normal data set, showing the characteristic bell shape.



**Figure 1.7:** A histogram of a data set that is approximately normal. While not perfectly symmetric, it follows the general bell shape.

Many datasets observed in practice, from human height to measurement errors, follow this specific shape. Such data is said to be **normally distributed**.

#### Definition 1.15 (Normal Distribution)

A **normal distribution** is a symmetric, bell-shaped distribution where the mean, median, and mode are all equal and located at the center. Its shape is determined entirely by its mean ( $\mu$ ) and standard deviation ( $\sigma$ ).



When the bins of a histogram become smaller, the histogram becomes more and more similar to a normal distribution. This is because the normal distribution is a continuous distribution, while the histogram is a discrete distribution. This is a consequence of the **Law of Large Numbers**.

#### Theorem 1.1 (Law of Large Numbers)

The sample mean of a dataset will converge to the population mean as the sample size increases.



When the sample size is large enough, the sample mean will be very close to the population mean, and the histogram will be very similar to a normal distribution and will get the following shape:

For data that is approximately normal, we can use a powerful rule of thumb to understand its spread without looking at every single data point. This is known as the **Empirical Rule**.

#### Theorem 1.2 (The Empirical Rule)

If a dataset is approximately normal with a sample mean  $\bar{x}$  and sample standard deviation  $s$ , then:

- Approximately **68%** of the observations lie within 1 standard deviation of the mean ( $\bar{x} \pm s$ ).

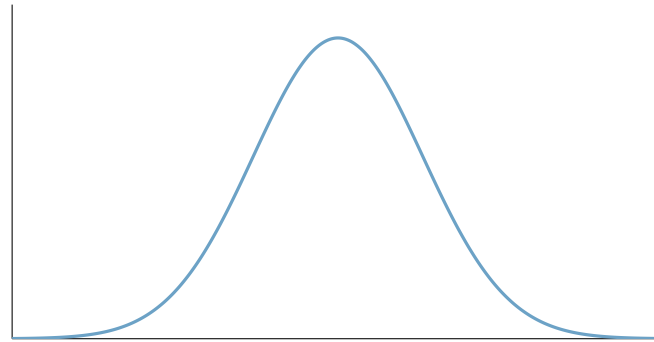



Figure 1.8: The normal distribution curve.

- Approximately **95%** of the observations lie within 2 standard deviations of the mean ( $\bar{x} \pm 2s$ ).
- Approximately **99.7%** of the observations lie within 3 standard deviations of the mean ( $\bar{x} \pm 3s$ ). 

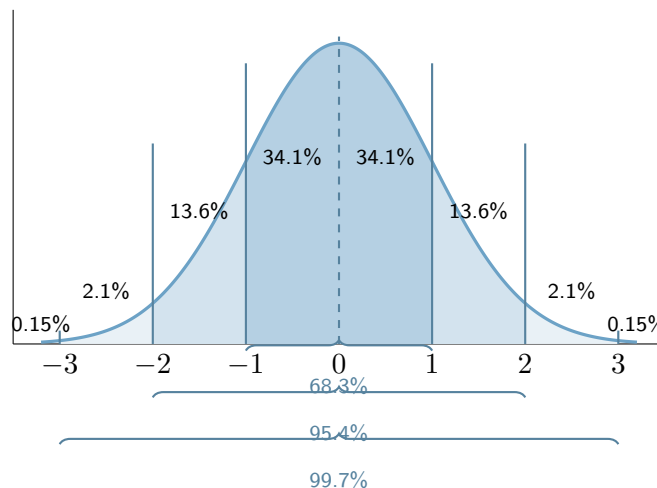


Figure 1.9: Empirical Rule on a normal curve using 'mseViaBlue'. The inner bands mark  $\pm 1\sigma$  (68.3%),  $\pm 2\sigma$  (95.4%), and  $\pm 3\sigma$  (99.7%).

## Skewed Data Sets

When a data set is not symmetric about its sample median, it is described as **skewed**. A long tail extending to the right indicates the distribution is **skewed to the right**; a long tail to the left means it is **skewed to the left**. Thus, the histogram in Figure 1.10 is skewed left, while the one in Figure 1.11 is skewed right.

### Definition 1.16 (Skewness)

**Skewness** measures the asymmetry of data distribution:

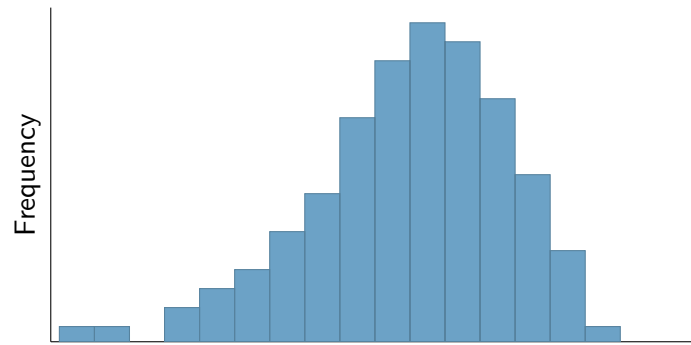
- **Positive skew (skewed to the right)**: Tail extends to the right (mean  $>$  median)
- **Negative skew (skewed to the left)**: Tail extends to the left (mean  $<$  median)
- **Symmetric**: Mean  $\approx$  median



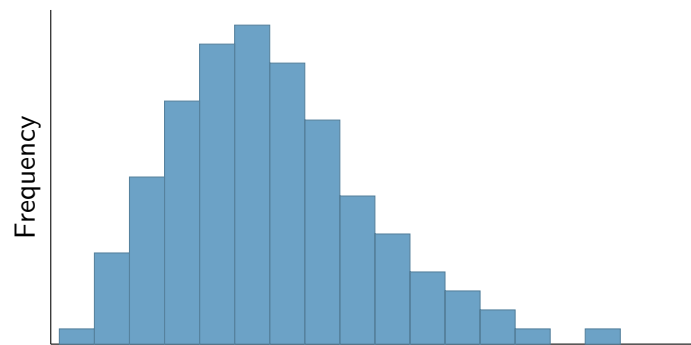
## Box Plots

Box plots provide a compact summary of data distribution, highlighting the median, quartiles, and outliers.

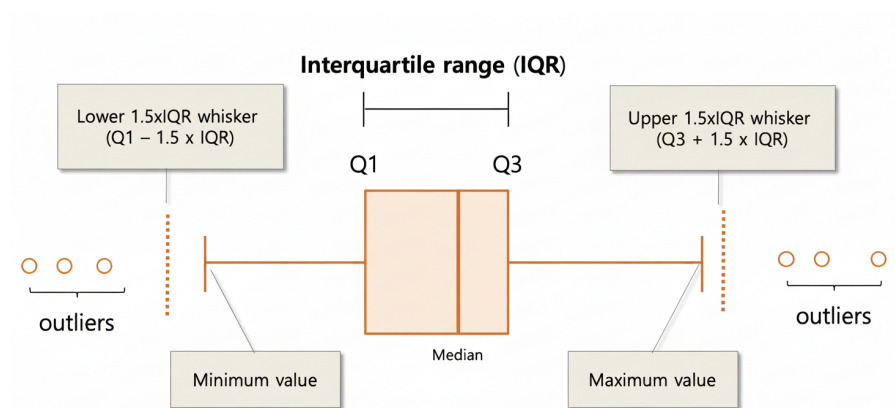
Figure 1.12 shows an example of a box plot.



**Figure 1.10:** A histogram of a data set that is skewed to the left (negatively skewed). The tail of the distribution is on the left.



**Figure 1.11:** A histogram of a data set that is skewed to the right (positively skewed). The tail of the distribution is on the right.



**Figure 1.12:** An example of a box plot, illustrating the median, quartiles, whiskers, and potential outliers.

#### Definition 1.17 (Box Plot Components)

A box plot displays five key statistics:

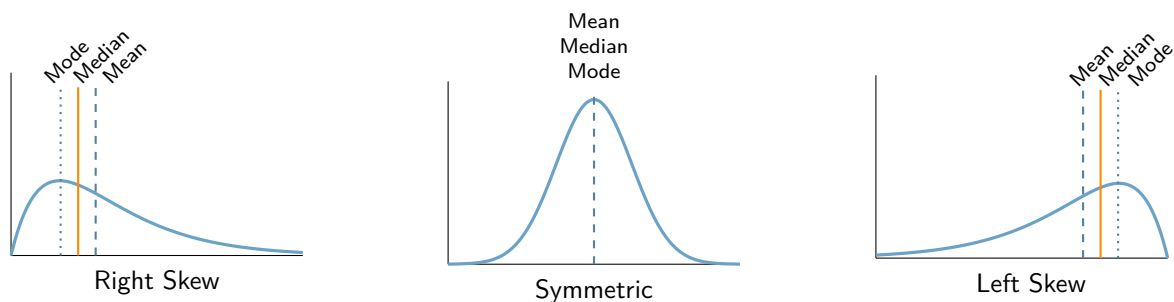
- **Minimum:** The smallest value (excluding outliers)
- **Q1 (First Quartile):** 25% of data below this value
- **Median (Q2):** 50% of data below this value
- **Q3 (Third Quartile):** 75% of data below this value
- **Maximum:** The largest value (excluding outliers)
- **Outliers:** Points beyond  $1.5 \times IQR$  from the box



A box plot provides a visual summary of the distribution of a dataset, making it easy to compare groups

and spot outliers. Here's how to read a box plot.

- The **length of the box** shows the spread of the central half of the data. A longer box means more variability in the middle 50%.
- The **position of the median line** within the box indicates skewness. If the median is closer to  $Q_1$  of the box, the data are skewed right; if closer to  $Q_3$  of the box, skewed left.
- The **position of the mean** can be determined by the position of the median line. If the median line is closer to  $Q_1$  of the box, the mean is to the left of the median; if closer to  $Q_3$  of the box, the mean is to the right of the median.
- The **position of the mode** can also be determined by the position of the median line. If the median line is closer to  $Q_1$  of the box, the mode is to the left of the median; if closer to  $Q_3$  of the box, the mode is to the right of the median.
- The **whiskers** show the range of the bulk of the data, excluding outliers.
- **Outliers** highlight unusually high or low values that may warrant further investigation.



**Figure 1.13:** Relationship between mean, median, and mode for positively skewed, symmetric, and negatively skewed distributions. Curves and annotations use the book's color 'mseViaBlue'.

Box plots are especially useful for comparing distributions across several groups or datasets, as they quickly reveal differences in medians, spreads, and the presence of outliers.

This concludes our discussion of data visualization and summary statistics, and in general our discussion of probability and statistics.

## Appendix A: Important Concepts

This appendix is a collection of important mathematical concepts that are frequently used in software engineering. The content of this appendix is based on the concepts in this book.

### Proposition A.1 (Order of Operations)

To evaluate mathematical expressions, operations are performed in the following order:

1. **Brackets (Parentheses):** First, perform all operations inside brackets or parentheses.
2. **Exponents and Radicals:** Next, evaluate exponents (powers) and radicals (roots).
3. **Multiplication and Division:** Then, perform multiplication and division from left to right.
4. **Addition and Subtraction:** Finally, execute addition and subtraction from left to right.



### Proposition A.2 (Rules for Calculations with Fractions)

For  $a, b, c, m \in \mathbb{R}$ , with  $a, b, c, m \neq 0$  where required, the following identities hold:

$$(1) \quad \frac{a}{b} \times m = \frac{am}{b}$$

$$(2) \quad \frac{a}{b} \div m = \frac{a}{bm}$$

$$(3) \quad m \div \frac{a}{b} = \frac{mb}{a}$$

$$(4) \quad \frac{a}{b} \times \frac{c}{a} = \frac{c}{b}$$

$$(5) \quad \frac{a}{b} \div \frac{c}{a} = \frac{a^2}{bc}$$

$$(6) \quad \frac{a}{b} = \frac{ac}{bc}$$

$$(7) \quad \frac{a}{b} + \frac{c}{a} = \frac{a^2 + bc}{ab}$$



### Proposition A.3 (Properties of Integer Exponents)

Let  $n, m \in \mathbb{Z}$ . Then the following hold (with  $x, y \in \mathbb{R}$  and nonzero where stated):

$$(1) \quad x^n \cdot x^m = x^{n+m},$$

$$(2) \quad \frac{x^n}{x^m} = x^{n-m} \quad \text{with } x \neq 0,$$

$$(3) \quad x^n \cdot y^n = (xy)^n,$$

$$(4) \quad \frac{x^n}{y^n} = \left(\frac{x}{y}\right)^n \quad \text{with } y \neq 0,$$

$$(5) \quad (x^n)^m = x^{nm},$$

$$(6) \quad x^1 = x.$$



### Proposition A.4 (More Properties of Integer Exponents)

Let  $n, m \in \mathbb{Z}$ . Then the following hold (with  $x, y \in \mathbb{R}$  and nonzero where stated):

$$(7) \quad x^0 = 1 \quad x \neq 0$$

$$(8) \quad \frac{1}{x^m} = x^{-m} \quad x \neq 0$$



### Rules for rearranging formulae

The following operations can be performed on both sides of the formula:

- Add the same quantity to both sides
- Subtract the same quantity from both sides
- Multiply both sides by the same quantity - remember to multiply all terms
- Divide both sides by the same quantity - remember to divide all terms
- Apply a function to both sides, such as squaring or finding the reciprocal

### Definition A.1 (Injective and Surjective Functions)

A function  $f : A \rightarrow B$  is called **one-to-one** (or **injective**) if different elements in  $A$  map to different elements in  $B$ . A function  $f : A \rightarrow B$  is called **onto** (or **surjective**) if every element in  $B$  is the image of at least one element in  $A$ .



### Definition A.2 (Inverse Functions)

Let  $f$  be a one-to-one correspondence from the set  $A$  to the set  $B$ . The inverse function of  $f$  is the function that assigns to an element  $b$  belonging to  $B$  the unique element  $a$  in  $A$  such that  $f(a) = b$ . The inverse function of  $f$  is denoted by  $f^{-1}$ . Hence,  $f^{-1}(b) = a$  when  $f(a) = b$ .

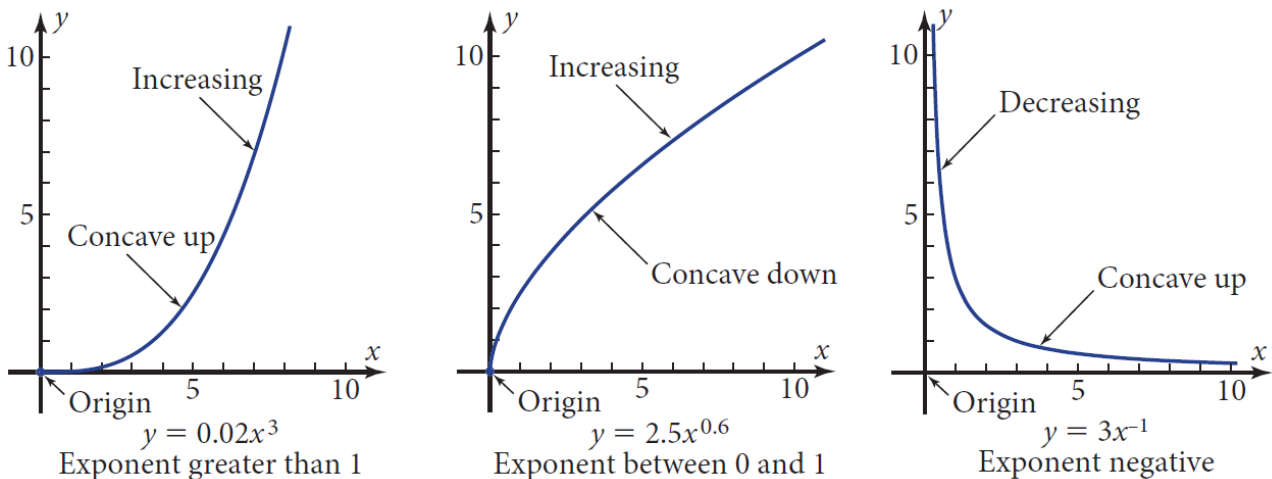


Figure A.1: Power functions

### Definition A.3 (Base-10 Logarithms)

$$\log x = y \iff 10^y = x$$

Verbally:  $\log x$  is the exponent in the power of 10 that gives  $x$



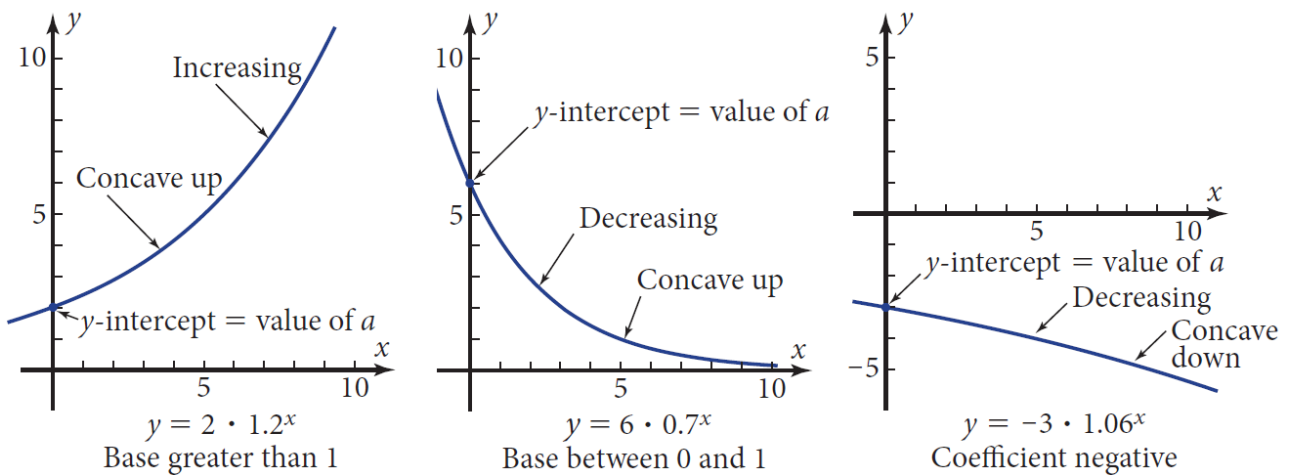


Figure A.2: Exponential functions

### Properties of base-10 logarithms

- Log of a Product:

$$\log xy = \log x + \log y$$

*Verbally:* The log of a product equals the sum of the logs of the factors.

- Log of a Quotient:

$$\log \frac{x}{y} = \log x - \log y$$

*Verbally:* The log of a quotient equals the log of the numerator minus the log of the denominator.

- Log of a Power:

$$\log x^y = y \log x$$

*Verbally:* The log of a power equals the exponent times the log of the base.

### Definition A.4 (Common Logarithm and Natural Logarithm)

*Common:* The symbol  $\log x$  means  $\log_{10} x$ .

*Natural:* The symbol  $\ln x$  means  $\log_e x$ , where  $e$  is a constant equal to 2.71828182845...

### The Change-of-Base Property of Logarithms

$$\log_a x = \frac{\log_b x}{\log_b a} \quad \text{or} \quad \log_a x = \frac{1}{\log_b a} (\log_b x)$$



### Properties of Logarithms

The Logarithm of a Power:

$$\log_b x^y = y \log_b x$$

The Logarithm of a Product:

$$\log_b(xy) = \log_b x + \log_b y$$

The Logarithm of a Quotient:

$$\log_b \frac{x}{y} = \log_b x - \log_b y$$